

CLAIMS

1 1. A system for controlling co-scheduling of processes in a computer comprising at least one process
2 and a spin daemon, the process being configured to, when it is waiting for a flag to change condition,
3 transmit a flag monitor request to the spin daemon and de-schedule itself, the spin daemon being
4 configured to, after receiving a flag monitor request monitor the flag and, after the flag changes
5 condition, enable the at least one process to be re-scheduled for execution by the computer.

2 2. A system as defined in claim 1 in which said spin daemon is configured to monitor a plurality of
3 flags, each in response to a flag monitor request, the spin daemon maintaining a list identifying those
4 flags it is to monitor, the spin daemon being further configured to, when it receives a flag monitor
5 request, add an identification of a flag associated with the request to the list.

3 3. A system as defined in claim 2 in which said flags are contained in a memory segment, the spin
4 daemon being configured to enable the at least one process to be re-scheduled following a change
5 of condition of any flag in said memory segment.

1 4. A system as defined in claim 1 in which said at least one process is configured to register with
2 said spin daemon, during registration the at least one process being configured to provide the spin
3 daemon with an identifier for the memory segment, the spin daemon being configured to provide a
4 handle, the at least one process being configured to use the handle in the flag monitor request.

-16-

1 5. A system as defined in claim 1 in which said at least one process and said spin daemon are
2 configured to communicate over a socket.

1 6. A method of controlling co-scheduling of processes in a computer comprising at least one process
2 and a spin daemon, the method comprising the steps of:

3 A. enabling the process to, when it is waiting for a flag to change condition, transmit a flag
4 monitor request to the spin daemon and de-schedule itself,

5 B. enabling the spin daemon to, after receiving a flag monitor request monitor the flag and, after
6 the flag changes condition, enable the at least one process to be re-scheduled for execution
7 by the computer.

8 7. A method as defined in claim 6, the spin daemon being configured to monitor a plurality of flags,
9 each in response to a flag monitor request, the spin daemon maintaining a list identifying those flags
10 it is to monitor, the method including the step of enabling the spin daemon being to, when it receives
11 a flag monitor request, add an identification of a flag associated with the request to the list.

1 8. A method as defined in claim 7 in which said flags are contained in a memory segment, the
2 method including the step of enabling the spin daemon to enable the at least one process to be re-
3 scheduled following a change of condition of any flag in said memory segment.

1 9. A method as defined in claim 6 further including the steps of

- 2 A enabling the at least one process to register with said spin daemon, during registration the
3 at least one process being configured to provide the spin daemon with an identifier for the
4 memory segment; and
- 5 B. enabling the spin daemon to provide a handle for use by the at least one process in the flag
6 monitor request.

1 10. A method as defined in claim 6 further comprising the step of enabling the at least one process
2 and said spin daemon communicate over a socket.

3 11. A computer program product for use in connection with a computer to control co-scheduling of
4 at least one process in the computer, the computer program product including a computer readable
5 medium having encoded thereon:

- 6 A. a process module configured to enable the computer to, when the process is waiting for a flag
7 to change condition, transmit a flag monitor request to the spin daemon and de-schedule
8 itself,
- 9 B. a spin daemon module configured to enable the computer to, after receiving a flag monitor
request monitor the flag and, after the flag changes condition, enable the at least one process
to be re-scheduled for execution by the computer.

1 12. A computer program product as defined in claim 11 in which said spin daemon is configured to
2 enable the computer to monitor a plurality of flags, each in response to a flag monitor request, the
3 spin daemon enabling the computer to maintain a list identifying those flags it is to monitor, the spin

-18-

4 daemon being further configured to enable the computer to, when it receives a flag monitor request,
5 add an identification of a flag associated with the request to the list.

1 13. A computer program product as defined in claim 12 in which said flags are contained in a
2 memory segment, the spin daemon being configured to enable the computer enable the at least one
3 process to be re-scheduled following a change of condition of any flag in said memory segment.

1 14. A cp as defined in claim 11 in which said at least one process is configured to enable the
2 computer to register with said spin daemon, during registration the at least one process being
3 configured to enable the computer to provide the spin daemon with an identifier for the memory
4 segment, the spin daemon being configured to enable the computer to provide a handle, the at least
5 one process being configured to use the handle in the flag monitor request.

1 15. A computer program product as defined in claim 11 in which said at least one process and said
2 spin daemon are configured to enable the computer to communicate over a socket.